CISC 106 Summer 2014 Project – Part I

Submission Deadline: 07/30/2014 11:55pm

* Form teams of two people. If you are not willing to work in a group, you can work by yourself.

* Write at least 3 test cases for each function you implement. If the function gets user input, you do not need to write test cases.

* Write both team members name at the top of project files.

In this project you are going to implement a version of the battleship game. If you do not know the rules of the game, read the following Wikipedia page about the game (<u>http://en.wikipedia.org/wiki/Battleship_(game</u>)). You can also play the online version of the battleship game at <u>http://www.battleshiponline.org/</u>.

The game that you will develop is going to be a little more complicated than the regular battleship game, but the main logic is going to be same. So, before continuing to read this document, first make sure you know how to play the battleship game and spend some time thinking about how you would design the classic battleship game. Let me give you a hint. When I played the battleship at <u>http://www.battleshiponline.org/</u> the program first showed me a grid board which has two pieces; one piece for me and one piece for my opponent (a computer) and asked me to place my ships to the board. Then the game continued with firing each other's board on a turn-by-turn basis. When there was no ship left on my opponent's board, the computer informed me that I won. So what I noticed is that the game has 3 stages which are: creating and initializing the boards, inserting the ships, and playing the game in turns. You can assume that the game you are going to develop will have same stages.

You should first think about how you are going to represent the two boards (one for first user and one for second user) in your program and how you are going to simulate the turn shifts before embarking on coding. A simple solution to representing boards and turn shifts can be using two dimensional nested lists (for boards) and loops (for turn shifts).

CISC 106 Battleship Game Requirements



Board creation and initialization:

- Battleship is a board game so your program should create a board for each of the two players. You may choose to use a two dimensional nested list for board data structure.
- Each board should be 20 cells wide and 20 cells long.
- Boards should be initialized with '~' character.

Ship Insertion

• After board creation and initialization the program should ask users to insert their ships to their boards' one ship at a time. After each insertion operation the program should display the updated board. The program should start to ask to second user to insert ships after the first user completes the insertion operations. One thing that you have to be careful of with this operation is that the opponents' ship locations should not visible to the each other. You can use the **cls** function below after each turn shift for this operation.

```
def cls():
print("\n" * 100)
```

- During an insertion operation of a ship the program should ask the user the following questions; (1) what coordinate values you want to use for your ship, (2) what direction you want to place/point your ship in. The values that the user enters (an x and a y coordinate value) are going to be the center of the ship. There are only two directions for the ship that can be either vertical ('V') or horizontal ('H').
- Each navy consists of two battleships, a destroyer, a frigate and a patrol ship. Each ship should be represented with a character on the board. Use 'B' for battleship, 'D' for destroyer, 'F' for frigate and 'P' for patrol ship.
- A battleship is 7 cells long and it has 3 guns.
- A destroyer is 5 cells long, has 2 guns.
- A frigate is 5 cells long, has 2 guns
- A patrol ship is 3 cells long, has 1 gun.

In the picture below you can see a board output

			YOUR BOARD															TARGET BOARD																						
0	1	02	03	04	05	06	07	80	09	10	11	12	13	14	15	16	17	18	19	20	01	02	03	04	05	06	07	80	09	10	11	12	13	14	15	16	17	18	19	20
01	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
02	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
03	~	~	D	D	D	D	D	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
04	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
05	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
06	~	~	~	~	~	~	~	~	~	~	~	в	в	в	в	в	в	в	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
07	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0.8	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
09	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
10	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
11	_	_	_	_	_	_	_	TP.		T.			_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
12	2							-	-	-		-																												
12			-	-	-	-	-	-		-	-	-	-	-	-		-	-	-	-	-		-	-		-	-	-		-	-	-	-	-	-	-	-	-	-	
13	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
14	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
15	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
16	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	Р	P	P	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
17	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
18	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
19	~	в	в	в	в	в	в	в	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
20	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~